

Technical Report
CMU/SEI-89-TR-3
ESD-TR-89-3

2



Carnegie-Mellon University
Software Engineering Institute

AD-A227 426

DTIC FILE COPY

The Role of Assessment in Software Process Improvement

David H. Kitson
Watts S. Humphrey

December 1989

DTIC
ELECTE
OCT 05 1990
S B D
CD

REPRODUCED BY
U.S. DEPARTMENT OF COMMERCE
NATIONAL TECHNICAL
INFORMATION SERVICE
SPRINGFIELD, VA 22161

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

90 10-01-057

The following statement of assurance is more than a statement required to comply with the federal law. This is a sincere statement by the university to assure that all people are included in the diversity which makes Carnegie Mellon an exciting place. Carnegie Mellon wishes to include people without regard to race, color, national origin, sex, handicap, religion, creed, ancestry, belief, age, veteran status or sexual orientation.

Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admissions and employment on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders. In addition, Carnegie Mellon does not discriminate in admissions and employment on the basis of religion, creed, ancestry, belief, age, veteran status or sexual orientation in violation of any federal, state, or local laws or executive orders. Inquiries concerning application of this policy should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

THE ROLE OF ASSESSMENT IN SOFTWARE PROCESS
IMPROVEMENT

Carnegie-Mellon University
Pittsburgh, PA

Dec 89

NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE BEST
COPY FURNISHED US BY THE SPONSORING AGENCY.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS NONE		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE DISTRIBUTION UNLIMITED		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-TR-89-3			5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-89-3		
6a. NAME OF PERFORMING ORGANIZATION SOFTWARE ENGINEERING INST.		6b. OFFICE SYMBOL (If applicable) SEI		7a. NAME OF MONITORING ORGANIZATION SEI JOINT PROGRAM OFFICE	
6c. ADDRESS (City, State and ZIP Code) CARNEGIE-MELLON UNIVERSITY PITTSBURGH, PA 15213			7b. ADDRESS (City, State and ZIP Code) ESD/XRS1 HANS COM AIR FORCE BASE HANS COM, MA 01731		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION SEI JOINT PROGRAM OFFICE		8b. OFFICE SYMBOL (If applicable) ESD/XRS1		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962885C0003	
8c. ADDRESS (City, State and ZIP Code) CARNEGIE-MELLON UNIVERSITY PITTSBURGH, PA 15213			10. SOURCE OF FUNDING NOS.		
			PROGRAM ELEMENT NO. 63752F	PROJECT NO. N/A	TASK NO. N/A
					WORK UNIT NO. N/A
11. TITLE (Include Security Classification) The Role of Assessment in Software Process Improvement					
12. PERSONAL AUTHOR(S) David H. Kitson and Watts S. Humphrey					
13a. TYPE OF REPORT FINAL		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Yr., Mo., Day) December 1989	
				15. PAGE COUNT 40	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB GR	assessments software engineering		
			maturity model software process improvement		
			SEI assessments software process management		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) This report discusses the role of assessment in improving an organization's software capabilities; specifically, the ability of the organization's projects to consistently meet cost, schedule, and quality objectives. Software process assessments are described from both a conceptual and pragmatic point of view. Underlying concepts of software process, software process management, and software process maturity are discussed. Collectively, these constitute a framework for software process assessment and improvement.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input checked="" type="checkbox"/>			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED, UNLIMITED DISTRIBUTION		
22a. NAME OF RESPONSIBLE INDIVIDUAL KARL H. SHINGLER			22b. TELEPHONE NUMBER (Include Area Code) 412 268-7630		22c. OFFICE SYMBOL SEI JPO

Technical Report

CMU/SEI-89-TR-3

ESD-TR-89-3

December 1989

**The Role of Assessment
in Software Process Improvement**



David H. Kitson

Watts S. Humphrey

Software Process Assessment Project

Approved for public release.
Distribution unlimited

Software Engineering Institute

Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

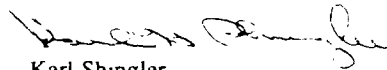
SEI Joint Program Office
ESD/AVS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER



Karl Shingler
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

Copyright © 1989 Carnegie Mellon University

This document is available through the Defense Technical Information Center. DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145.

Copies of this document are also available through the National Technical Information Service. For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Table of Contents

1. Software Process Overview	1
1.1. Introduction	1
1.2. The State of Practice of Software Engineering	2
1.3. The Process View	3
1.4. Role of the Software Process	3
1.5. Benefits of a Good Process	5
1.6. Process in Perspective	5
2. Principles of Software Process Management	7
2.1. The Discipline of Software Process Management	7
2.2. Process Quality	8
2.3. Statistical Control	9
2.4. Management Responsibility	10
2.5. Process Definition	10
2.6. Process Support	11
3. Introduction to Software Process Assessment	13
3.1. A Paradigm for Software Process Improvement	13
3.2. SEI Software Process Maturity Framework	14
3.3. Assessment Defined	15
3.4. Assessment Principles	17
3.4.1. Software Process Framework	17
3.4.2. Strict Confidentiality	18
3.4.3. Sponsorship	19
3.4.4. Attitude and Teamwork	19
3.4.5. Action Focus	20
3.5. Assessment Ground Rules	21
3.6. Implementation Risks	21
4. Conducting Software Process Assessments	23
4.1. Assessment Phases	23
4.2. Preparing for an Assessment	23
4.2.1. Forming an Assessment Team	23
4.2.2. Preparing the Assessment Team	24
4.2.3. Preparing the Site	26
4.2.3.1. Spreading the Word	26
4.2.3.2. Selecting Projects	26
4.2.3.3. Selecting Functional Area Representatives	27
4.2.3.4. Preparing Participants	27
4.3. Conducting the Assessment	28
4.3.1. Opening Assessment Briefings	29
4.3.2. Reviewing Project Responses	29
4.3.3. Assessment Discussions	29

4.3.3.1. General Considerations	29
4.3.3.2. Discussions with Project Leaders	30
4.3.3.3. Discussions with Functional Area Representatives	31
4.3.4. Formulating Findings	31
4.3.5. Presenting Findings	31
4.4. Recommendations	32
4.4.1. Considerations in Formulating Action Recommendations	32
4.4.2. The Written Assessment Report	32
4.4.2.1. Role of the Report	32
4.4.2.2. Content of the Report	33
4.4.3. Recommendations Briefing	33
5. Establishing an Assessment Program	34
5.1. Establishing a Corporate Assessment Program	34
5.2. Planning a Corporate Assessment Program	35
5.3. Staffing for Corporate Assessment Groups	35
5.4. Initiating Corporate Assessments	36
5.5. Establishing a Site Assessment Program	36
5.6. Other Considerations	37
 References	 38

List of Figures

Figure 2-1:	The Software Process Engineering View	8
Figure 3-1:	A Software Process Improvement Cycle	14
Figure 3-2:	SEI Software Process Maturity Model	15
Figure 4-1:	SEI On-Site Assessment Process Flow	28

Accession For	
NTIS SPAN	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Acknowledgments

The authors wish to express appreciation for the efforts of all the individuals and sponsoring organizations that were involved in the various software process assessments, and to our colleagues at the SEI who reviewed the drafts that led to this report. We particularly appreciate the efforts of the review team: Rodger Blair, Anita Carleton, Louise Hawthorne, and Tim Kasse. Also, we acknowledge Linda Pesante, who significantly improved the readability of the report through her technical editing skills.

The Role of Assessment in Software Process Improvement

Abstract: This report discusses the role of assessment in improving an organization's software capabilities; specifically, the ability of the organization's projects to consistently meet cost, schedule, and quality objectives. Software process assessments are described from both a conceptual and pragmatic point of view. Underlying concepts of software process, software process management, and software process maturity are discussed. Collectively, these constitute a framework for software process assessment and improvement.

1. Software Process Overview

1.1. Introduction

Our focus in this report is the ability of software organizations to produce (and evolve) software systems within the constraints of cost, schedule, and quality. We contend that successful software suppliers *will increasingly require* software development¹ processes that are explicitly defined, measured, and managed.

The report is organized into five chapters. The first two provide a conceptual foundation for the remaining three, which discuss the principles and practice of software process assessment.

Chapter 1 introduces the notion of software process, discusses its role in relation to people and technology, and provides the motivation for focusing on the software process.

Chapter 2 introduces software process management and discusses some of its fundamental principles.

Chapter 3 provides an overview of software process assessment, introduces a software process improvement paradigm, and discusses the underlying principles and implementation risks of the assessment process.

Chapter 4 describes how assessments are conducted. Assessments conducted by the Software Engineering Institute (SEI) are used as the basis for this discussion.

¹Unless otherwise indicated, we use the term development in its broadest sense to include the activities traditionally labeled *maintenance*.

Finally, Chapter 5 discusses how organizations can establish their own assessment programs.

1.2. The State of Practice of Software Engineering

Much current software practice is reminiscent of how Rita Mae Brown, an American poet, defines insanity: doing the same thing over and over, and expecting different results. Ongoing work at the Software Engineering Institute to characterize the state of software engineering practice in the Department of Defense (DoD) software community indicates that the majority of software organizations are operating at an immature level² of software process capability [HUM89]. In a mature software process, an organization combines methods, techniques, and technology to produce consistent results. In an immature software process, costs and schedules are largely unpredictable, quality is generally marginal, and technology is often used ineffectively. Specifically, organizations with immature processes are deficient in one or more of the following areas:

- Project planning
- Project management
- Configuration management
- Software quality assurance

Assessments of several dozen large software organizations (many of which were conducted by the SEI) make it increasingly clear that the organizations all face similar problems [HUM89]. What is more, these problems have all been solved before, and often in the same organization!

Software professionals generally need the most help in controlling requirements, coordinating changes, managing (and making) plans, managing interdependencies, and dealing with systems design issues. Since these and similar problems consume a large part of every programmer's time, this is where management can provide the most immediate help. The surprising thing is, at least for low maturity organizations, technical issues rarely appear at the top of priority lists. This is not because technical issues are not important; it is because so many management problems must be handled first.

A mature software process does not eliminate the need to understand the application, to deal with changing requirements, and to manage system design issues. However, organizations with more mature processes are better positioned to address the issues effectively and avoid the unnecessary exacerbation of these and other more mundane problems.

²The concept of software process maturity is discussed in Section 3.2.

Since critical defense systems are becoming increasingly reliant on complex software, aggressive improvement actions are required.

1.3. The Process View

Brooks has pointed out that there is no magical silver bullet that will solve all the problems, at least not in the foreseeable future [BRO87]. In light of this, our approach has been to take a process perspective in considering industrial efforts to produce software within the constraints of cost, schedule, and quality. This approach treats the way software professionals produce software as a separate, distinct entity that can be described, defined, studied, measured, managed, and improved. The motivation for this perspective is a basic principle of industrial engineering:³ the quality of a product is governed by the quality of the process used to develop and evolve it.

A process is a set of activities, methods, and practices that guide people (with their tools) in the production of goods or services. The software process is that process used to develop or evolve software products. A fully effective software process must consider the relationships of the required tasks, the tools and methods, and the skill, training, and motivation of the people involved.

Software organizations employ a software process regardless of whether it is explicitly defined, documented, and managed. Every software organization has, as a minimum, the de facto process--the state of practice among its current software professionals. Typically, there is little conscious attention given to the de facto software process; it just happens. So it is important to understand that the issue is *not* whether a software organization uses a software process, but whether it *manages* the software process it already has.

1.4. Role of the Software Process

There are several aspects to the role played by a well-defined and effective software process. First, a defined and documented software process provides a framework for the key activities of software development. The role is akin to roadways and traffic laws. Imagine the chaos resulting from a lack of established roadway systems, laws, and enforcement mechanisms. This characterization of the state of many software organizations is not altogether unfair.

Secondly, the software process provides a vehicle for defining expectations for key activities in terms of input and output criteria. Because the production of software is complex, the people involved frequently find it difficult to fully understand why a process has been designed in a particular way and, more

³Software development is, at least in part, an industrial engineering activity [BAU75].

importantly, what the implications of deviating from the defined process are. Defined software processes make explicit the interdependencies of activities, thus directly supporting the producer-consumer paradigm⁴ within the software organization.

Finally, a process focus helps to assure that the organization benefits from experience by providing the equivalent of culture or tradition either to help avoid similar problems or to successfully deal with them. With time, the software process evolves, grows, and improves as it responds to and deals with new situations and challenges.

In commenting on his successes in physics, Sir Isaac Newton said:

*If I have seen further than those who proceeded me, it is because I have stood on the shoulders of giants.*⁵

Although Sir Isaac may have been standing on the shoulders of giants, all too often today's software professionals are standing on each others' toes! With a mature software process, software organizations will increasingly be able to build on their experience and software professionals will be able to apply their skills to the most challenging technical problems.

1.5. Benefits of a Good Process

What are the benefits an organization can expect from explicitly defining, documenting, and managing its software process? Among the most important are the following:

- More appropriate and effective use of software professionals.
- A basis for quantitative software management.
- Sustained orderly improvement of the software process.

A frequently heard concern is that with so many rules and constraints (e.g., standards and policies), there will be no opportunities for creative and innovative software professionals. Often, the basis for this concern is confusion about what constitutes creative and innovative work. Typically, there are so many crises to handle in software projects that creative software people are consumed with such technically trivial problems as who made the last change to module X, and why!

⁴Producer-consumer paradigm: the view that each individual in the software development process is both a producer and consumer of information and other relevant artifacts.

⁵ Stephen Jay Gould, *The Panda's Thumb*, New York: W.W. Norton & Co., 1980, p.47.

An explicitly defined and documented software process provides the foundation for developing, over time, a quantitative software management capability. By this we mean the ability to make most management decisions on the basis of quantitative data characterizing the process and its effectiveness. In most organizations today, software management is largely intuitive. However, once a software organization has defined its software process and begins to understand and manage it quantitatively, it has achieved the capacity for sustained and orderly improvement. This will be vital for survival of the software businesses of the future.

A quantitative software management capability, for example, permits the organization to identify (in quantitative terms) the weaknesses in the process, their impact, and the potential gains from improvement actions. The ability to forecast, in quantitative terms, the potential return on investment in automation can greatly facilitate the approval process.

As an organization begins to augment its defined software process with metrics and gather and analyze process data, there will be a paradigm shift to management based on quantitative data.

1.6. Process in Perspective

In software engineering, people, process, and technology⁶ are mutually dependent on one another. All are critical components of an organization's software capability. A common but fallacious view is that one of these three is the *most* important. For example, historically, there has been a perception that software technology would be the silver bullet that would make the seemingly intractable problems collectively referred to as the *software crisis* go away. In fact, people, process, and technology share a relationship more like the legs of a triangle or the links in a chain. They are all important, and it is pointless to ask which is most important.

When a particular approach seems to fit a need, it is often tempting to assume that it will solve all the problems. Although process management provides a powerful basis for assessing software problems and a consistent framework for organizational improvement, it is not a cure-all. Having a mature software process does not guarantee success. Successful results can be produced by exceptional software professionals in spite of an immature process; however, from a business perspective, the risk involved is likely to be unacceptable.

Another key area that must be considered is the domain expertise of the application designers. In studying many software products to see what separated superior designs from others, Curtis found the successes were always designed by people who understood the application [CUR87,

⁶For the purpose of this discussion, we take software technology to mean software tools, methods, and environments.

CUR88]; for example, a well-designed program to control a missile was designed by someone who understood missiles. Convincing evidence indicates that superior products have superior designs. This may seem self-evident, but it is worth repeating. In that sense, a program can be viewed as executable knowledge. When application designers have domain knowledge coupled with the ability to produce a creative design, a high-quality product is likely to result. With such talents, an orderly process can be of great help. Without them, good product design is unlikely, regardless of the process used.

2. Principles of Software Process Management

2.1. The Discipline of Software Process Management

Software process management is the use of process engineering concepts, techniques, and practices to explicitly monitor, control, and improve the software process. The objective of software process management is to enable an organization to produce software products according to plan while simultaneously improving the quality of its products.

Many of the fundamental principles of process engineering and management are simple and straightforward. They have been applied in other industries, such as automobile manufacturing, chemical/pharmaceutical processing, and integrated circuit fabrication. The seminal work in process engineering and management was conducted during the first half of this century by Shewhart, Deming, Juran, and others [DEM82, WAL86].

The identification and characterization of these principles has been discussed in the literature [AGR81]. Card [CAR87] characterizes the discipline of process engineering and contrasts it with software engineering:

*To define this high-level conceptual and management approach, the underlying production process must be distinguished from day-to-day production activities. In manufacturing, the engineer who designs the product typically has different skills and responsibilities from the engineer who designs the factory in which the product is built. The same distinction should be made in software development. **Process engineering** views software development as a general production process distinct from any particular project. **Software engineering** is the application of this process within a project to develop a specific product.*

Card goes on to describe the high-level steps of production, product testing, and acceptance/operation, with the production step being the focus of process engineering. Figure 2-1, from his paper, illustrates graphically the view of process engineering discussed in that paper.

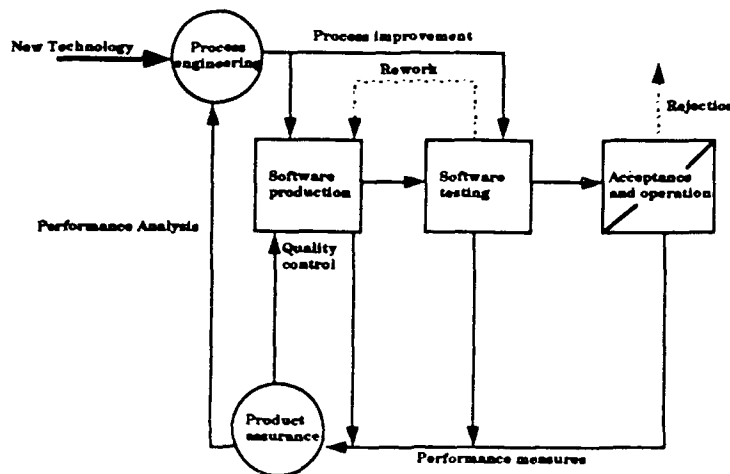


Figure 2-1: The Software Process Engineering View

The remainder of this chapter discusses the following principles of software process management:

1. The quality of a software product is governed by the quality of the process used to develop and evolve it.
2. Until the software process is under statistical control, orderly and sustained improvement is impossible.
3. The software process is a management responsibility.
4. The software process must be defined and documented.
5. The software process will not improve itself.

2.2. Process Quality

Many other industries have recognized that a defined, documented, and managed process is needed to ensure quality products. In traditional manufacturing of physical products such as automobiles and integrated circuits, this principle is self-evident and compelling. For example, no one would consider trying to fix chips at the end of the fabrication line. It is sobering to contemplate the amount of effort expended in the software industry by attempting to do essentially that with software systems.

How ironic it is that in the software industry, where the inner workings of our product can only be imagined, we have not yet fully recognized and accepted this principle. It is clearly necessary with such intangible artifacts that the process be the primary guarantor of product quality. While some testing will always be required, more emphasis is needed on process management and ultimately on process certification.

2.3. Statistical Control

When a process is under statistical control, repeating the work in roughly the same way will produce roughly the same result. All other factors being the same, it is thus necessary to improve the process to get consistently better results. Further, if the process is not under statistical control, sustained improvement is not possible.

W. Edward Deming, in his work with the Japanese after World War II, applied the concepts of statistical process control to many of their industries [DEM82]. While there are important differences between those industries and the development and evolution of software, many of the same concepts are as applicable to software as they are to automobiles, cameras, wrist watches, and steelmaking.

The basic principle behind statistical control is measurement. As Lord Kelvin said:

...when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the stage of science.

Processes that are under statistical control are amenable to examination using quantitative techniques such as control charts, which, among other things, provide quantitative guidelines for process "capacity." These, in turn, can help detect a particular process execution that has produced results which do not meet expectations.

Clearly, we are better off when our processes are under control. With quantitative guidelines, we can easily detect process steps where results do not meet expectations. But the significance of statistical process control goes beyond this. Until a process is under statistical control, it is impossible to know whether an intended improvement has had the expected impact on process output. Until this control has been achieved, one cannot tell whether variations in process performance are due to intrinsic causes or to the intended process improvement.

2.4. Management Responsibility

Perhaps Deming's most important contribution has been his insistence that process changes are management's responsibility [DEM82]. Most people will do the best job they can within the constraints of their working environment; exhortations to change cause little lasting improvement and often make things worse.

Changes to the software process must start at the top. Major changes require leadership and sponsorship. This requires a management team with the conviction that long-term improvement is both possible and essential. To produce results, these managers must insist on effective performance. Although they will not actually implement improvement actions, they must establish challenging goals, set the priorities, and furnish the resources. Moreover, sustained progress requires sustained management action. Although managers will not actually make the changes, they must provide continuing support; in addition to setting the priorities and furnishing resources, they must monitor progress and demonstrate their interest in process improvement.

2.5. Process Definition

One of the first improvement actions an organization must take is to define and document the existing process. A common understanding and agreement about the existing process enables groups of professionals and technicians to work together as a team. The definition of each process includes, as a minimum, the following information [RAD85]:

- Entry criteria: What are the preconditions for beginning this process? What should be true before work begins?
- Task description: What steps are needed to complete the task? Note that some of these steps will themselves be described in separate process definitions.
- Validation criteria: How will the adequacy of the work be determined? Applicable standards, adherence to operating instructions, etc., are typically cited here.
- Exit criteria: What are the postconditions for this process? What must be true before this work is considered complete? How has the "world" changed as a result of successfully completing the process?

2.6. Process Support

In most software organizations, no one works on improving the software process; everyone concentrates on projects and product delivery.

No one questions the need to design a manufacturing process before tools are ordered and production begins. Manufacturers must consider raw materials handling, design the process flow, select the tools, specify the controls, and oversee ordering, installation, and operation. The software process needs the same attention. If it is not designed, it will merely be adjusted to each successive crisis. Overall performance will be essentially unchanged, and a chaotic process will remain chaotic.

The SEI promotes the establishment of software engineering process groups (SEPGs) in software organizations. An SEPG is a group of software professionals that concentrates on improving the organization's software development process. The SEPG is a small but dedicated resource of competent and experienced professionals. Typically, 2 to 3 percent of the size of the organization is adequate. While some personnel should be permanently assigned to the SEPG, it is useful to rotate technical and management professionals into the SEPG for two- to three-year assignments. The role of the SEPG is to focus the process improvement effort. Its members lead assessments of the current operation and coordinate development of the resulting action plans. They are involved in action plan implementation and periodically report to management on progress.

An SEPG is chartered to facilitate the definition, documentation, and improvement of the organization's software process. Its ongoing functions include:

- Performing periodic software process assessments.
- Reporting status of the software process to senior management.
- Facilitating the definition and improvement of technical and management process.
- Facilitating the definition and maintenance of process standards.
- Establishing and maintaining a software process database.
- Initiating and providing process education and training.
- Identifying, screening, and evaluating appropriate candidate software technologies.
- Providing process consultation to software practitioners.

Once the SEPG is established, the group also actively participates in the action plan⁷ for process improvement by doing the following:

- Facilitating action plan development and review.
- Leading and coordinating action plan implementation.
- Establishing and monitoring pilot change efforts.
- Tracking the progress of action plan implementation against the plan.
- Conducting the periodic management reviews of the software process.

Additional areas of activity may be appropriate. These will depend on the specific findings and recommendations of the assessment team. Smooth transition and coordination between the action plan team and a newly established SEPG are necessary for effective implementation of the action plan.

Note that the SEPG function is not to be confused with the SQA function, which is an audit and enforcement mechanism. The SEPG works closely with and assists software projects by providing knowledge, guidance, and consultation on software technologies, methods, practices, and tools. The role of SQA is to enforce the current process while the SEPG is dedicated to changing it.

⁷ An action plan describes the improvement actions an assessed organization intends to carry out following a software process assessment. See Chapters 3 and 4 of this report for additional discussions of the role of action plans in process improvement.

3. Introduction to Software Process Assessment

Software process assessments help organizations characterize the current state of their software process. Well-run assessments also produce findings and recommendations which help organizations set objectives and priorities for process improvement. This chapter discusses a paradigm for facilitating software process improvements, a supporting SEI software process maturity framework, and some fundamental principles underlying assessments.

3.1. A Paradigm for Software Process Improvement

As discussed in the first two chapters, an important first step in addressing software problems is viewing the entire software task as a process that can be controlled, measured, and improved. To produce orderly improvement in their software capabilities, organizations must take the following steps:

1. Understand the current status of their software process.
2. Develop a vision of the desired software process.
3. Establish a list of required software process improvement actions in order of priority.
4. Produce a plan to accomplish these actions.
5. Commit the resources and execute the plan.
6. Start over at 1.

A specific implementation of this paradigm is used by the SEI to facilitate software process improvement in the DoD software community; it is shown in Figure 3-1.

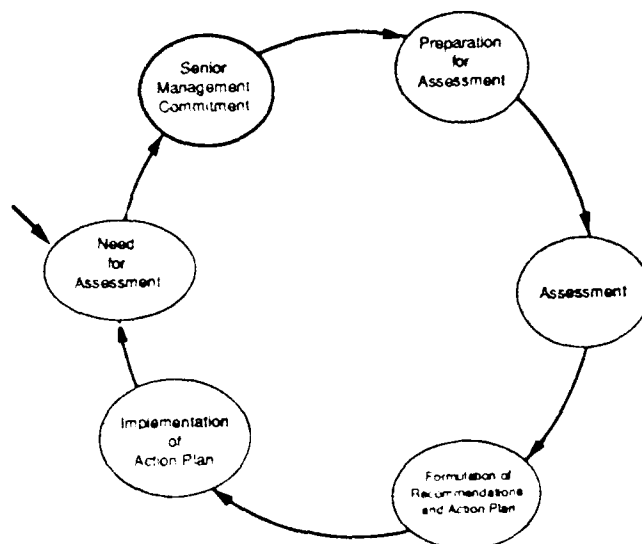


Figure 3-1: A Software Process Improvement Cycle

To actually improve an organization, it is helpful to have a clear picture of the ultimate goal and some way to gauge progress along the way. The framework used by the SEI characterizes the software process across five maturity levels. By establishing their organization's position in this framework, software professionals and their managers can more readily identify areas where improvement actions will be most fruitful. The next section briefly discusses this maturity framework, which is intended to be used with the assessment methodology described in the remainder of this report.

3.2. SEI Software Process Maturity Framework

As part of a continuing effort to aid the U.S. military services in identifying contractors with appropriate software capabilities, the SEI has developed a software process maturity framework (Figure 3-2) similar to Crosby's progressive management maturity grid [CRO79]. The maturity framework is an empirical model derived from the collective experiences of a number of experienced software managers and practitioners and is widely used by U.S. software organizations to guide their improvement efforts [HUM88].

Level	Characteristic	Key Problem Areas	Result
Optimizing	Improvement fed back into process	Automation	Productivity & Quality
Managed	(quantitative) Measured process	Changing technology Problem analysis Problem prevention	
Defined	(qualitative) Process defined and institutionalized	Process measurement Process analysis Quantitative quality plans	
Repeatable	(Intuitive) Process dependent on individuals	Training Technical practices • reviews, testing Process focus • standards, process groups	
Initial	(ad hoc / chaotic)	Project management Project planning Configuration management Software quality assurance	Risk

Figure 3-2: SEI Process Maturity Framework

These five levels have been selected because they:

- Reasonably represent the historical phases of evolutionary improvement of actual software organizations.
- Represent a measure of improvement that is reasonable to achieve from the prior level.
- Suggest interim improvement goals and progress measures.
- Make obvious a set of immediate improvement priorities once an organization's status in this framework is known.

While there are many aspects to an organization's transition from each maturity level to the next, the overall objective is to achieve a controlled and measured process. This provides the foundation for continuous process improvement.

3.3. Definition of Assessment

A software process assessment is an appraisal, or review, performed by a trained team of software professionals. Its purpose is to determine the current state of an organization's software process, to identify the highest priority process issues, and to facilitate improvement actions. A process assessment helps software organizations improve by identifying their critical software problems and establishing improvement priorities. The basic

objectives of an assessment are to learn how the organization works, identify its major problems, and enroll its opinion leaders in the change process [HUM87b].

John Gardner described the reason for doing assessments, saying that most organizations "are not suffering because they can't solve their problems but because they will not see their problems" [GAR65].

Management is often so focused on finding solutions that it fails to define the problems. Dale Zand, a professor at the New York University School of Business, notes that when managers say, "I don't want to hear your problems, I want to hear your solutions," they are taking the wrong approach [ZAN82]. At the other extreme, an unconstrained search for problems without regard to solutions rarely results in useful guidance. It is important to focus first on problem definition since complex problems must be thoroughly understood before a solution is attempted. When problems are ill-defined, the solutions are rarely useful; they may not even be pertinent to the actual problems faced by the professionals on a daily basis.

Software assessments are similar to the organizational development methods used successfully for many years [CON73, HUS75, ROD78]. They have also been used by both IBM and the SEI [HUM87a, HUM89, OLS89, RAD85]. The approach is to conduct a structured series of interviews with key people in the organization to learn their problems, concerns, and creative ideas.

Assessments differ from other studies that are commonly performed. Project reviews, for example, are generally used to identify the status of a particular project. Such evaluations are often initiated by senior management to probe specific issues or expose suspected problems. While this is a proper exercise of management responsibility, it is often a poor way to motivate change and generally provides little guidance on how to improve the software process.

Audits are also conducted for senior managers who suspect problems and send in experts to uncover them. In the financial field, examples of errors and occasional wrongdoing are so common that periodic financial audits are a sign of a well-run business. With software, periodic audits are also needed to maintain consistent focus on the way the work is supposed to be done. Some responsible engineering groups even make a practice of requesting audits of their own projects. Although this is not common, it can be very effective in helping these groups identify key issues. This practice is similar to the assessment process discussed in this report.

The main reason to audit software work, however, is to ensure that professionals follow the officially approved process. Based on our experience, typical deviations from the defined process are not motivated by greed but by a desire to get the job done as quickly and effectively as

practical. When professionals find that some aspects of the official process are outmoded and inefficient, they try to get the job done in spite of these bureaucratic obstacles, and their expedient shortcuts often turn out to be very effective. Thus, an audit can actually do more harm than good, particularly if the official process is not defined or cannot be implemented as stated.

Software audits can be highly effective when the software process is defined well enough to provide a standard. Comprehensive audits of large software organizations can be expensive, however, since they require a great deal of work by teams of skilled professionals.

3.4. Assessment Principles

A good assessment requires a competent team, sound leadership, and a cooperative organization. Because of the human-intensive nature of the software process, however, there are some special considerations:

- The need for a process model as a basis for the assessment.
- The requirement for confidentiality.
- Senior management involvement.
- Attitude and teamwork.
- An action orientation.

These points are described further in the following sections.

3.4.1. Software Process Framework

An assessment implies the existence of a standard: an organization's process is reviewed in comparison with some vision of how such processes should be performed. As the proverb says, "If you don't know where you are going, any road will do." The SEI has developed a process maturity model and assessment questionnaire to provide a foundation, or framework, for process assessments and evaluations conducted in the DoD software community [HUM88].

Without such a foundation, an assessment can easily become a loosely directed, intuitive exploration. If the assessment team members have extensive software experience and good intuition, such studies can still be valuable. However, when the members of such groups focus on their own particular specialties, the likely result is that no topics are covered in depth and many areas are overlooked. If such teams split into small units or assign individuals to probe particular areas, there is a better chance of covering all the key topics. Unfortunately, this approach may result in many different views of the operation, reducing the likelihood of a coherent result. Splitting the team also destroys the synergistic power of the group's diverse experience and minimizes the likelihood of agreement on anything but generalities.

These problems can be avoided when an assessment is based on a common view of the desired software process; this view provides a basis for orderly exploration as well as a framework for establishing problem priorities. With such a focus, the team can work together on the key issues and recommendations. While agreement may take some time, the discussions invariably stimulate deeper understanding, and far better conclusions are reached than would otherwise be possible.

3.4.2. Strict Confidentiality

The purpose of an assessment is to support the organization's improvement program and not to report its problems to higher management. Even when an assessment is initiated with this intent, it is extraordinarily difficult to maintain confidentiality, particularly when a chief executive demands to see the results. If any member of the assessment team provides such data, however, the organization's software professionals will learn that they cannot really speak in confidence. As this becomes widely known, the assessment group will find it increasingly difficult to conduct assessments that uncover the real issues.

Confidentiality permits the assessment team to talk to people at all levels of the organization. If the managers suspect that the findings will be passed to higher management, they will properly insist on being present at every interview. Unfortunately, when managers are present, professionals are unlikely to say anything that their managers don't know already know or with which they might disagree. There is then no reason to have an assessment; the managers could present this official view far more efficiently in a two-hour briefing.

Confidentiality is required at all organizational levels. The professionals must know that their comments will not be attributed to them. Several projects should be reviewed at once, and the project managers should be told that the results for their projects will be given only to them. Site management is then provided a composite picture of the overall operation. This ensures that no single project or individual is identified with any specific problem.

In short, both vertical (management) and horizontal (project) confidentiality is essential to ensure the free flow of information between the assessment team and the organization's software professionals.

3.4.3. Sponsorship

The senior manager sets the organization's priorities. This local manager typically gives final approval for software commitments and answers to corporate management when things go wrong. Although some senior managers are responsible for multiple projects in several locations, it is wise

to focus on a reasonably local geographic area. This minimizes project disruption, simplifies assessment arrangements, and generally facilitates subsequent action planning and implementation. In addition, it eases the assessment task by ensuring a reasonably homogeneous set of projects and software cultures. In the following discussions, the senior manager of this total organization will be called the site manager.

The site manager must be personally involved in the assessment and its follow-up action plans. If not, the work will not get sufficient priority. While some initial good-faith attempts may be made, experience has shown that the first crisis will soon preempt these process improvement efforts. To have any lasting impact, the site manager must personally participate, assign qualified people, and periodically review the progress of the resulting action plans.

Without this support, the assessment is likely to be a waste of time. The practitioners can generally handle their routine problems, but lasting improvements must survive periodic crises when the process is under most stress, when management is most likely to defer nonessential work, and when disasters are most likely. Since software crises are common, if the site manager does not protect the process improvement efforts, these efforts are not likely to continue long enough to bring about significant changes.

3.4.4. Attitude and Teamwork

A successful assessment is a collaborative effort to identify present good practice and key areas for improvement. An attitude of mutual respect between team members and all the other participants in an assessment is essential to the success of the effort.

One strength of the assessment process is that it is conducted as a structured study by a team of knowledgeable and experienced professionals. However, it is easy for an assessment to appear as an arrogant activity. A group of "experts" reviews a large and complex organization and in a few days tells them what they are doing wrong and what they should do to improve. Generally the organization's professionals work hard, are dedicated to doing a good job, and are trying to improve. They are thus properly skeptical of any brief study and doubt it can have any lasting impact.

The skepticism is not only understandable but is quite proper. A small team of experts cannot hope to identify in a few days the most critical problems in any organization. Complex problems rarely have simple answers, and the assessments are based on the assumption that an organization's software experts are in the best position to understand the software process issues. If an assessment team behaves as if it has all the answers, the others will soon sense it. Their natural reaction will be to show these "experts" they are not so smart after all, leading to an unspoken wish that the assessment will fail. Under these conditions, it often will.

With the leading in-house professionals contributing their knowledge and skills, the assessment can be a catalyst to motivate the entire organization to self-improvement. Thus, the assessment team learns during its training the importance of listening; team members guard against assuming they understand the problems before participants fully discuss them. Similarly, early briefings for participants encourage the view of assessments as a cooperative effort focused on learning and understanding. In addition, participants become aware that their contribution can help to change (and improve) the organization's software process.

Surprisingly, for each software problem, there is often someone in the organization who has already solved it. Making this capability visible can be one of the greatest and most immediate benefits of the assessment. Mistakes and oversights must be identified, but they are objectively reported without attribution, criticism, or blame. There must be sufficient rapport between the team and the organization's key people so the latter will share their problems, concerns, and creative ideas. Occasionally, a practitioner may be less than cooperative, even when the assessment team is appropriately supportive. If the team's actions clearly demonstrate their desire for active collaboration, however, this is recognized and people generally respond positively.

3.4.5. Action Focus

Finally, to have lasting effect, the assessment must be directed toward improvement. An action orientation keeps questions focused on current problems and the need to solve them. Otherwise, the assessment will not focus on the priority issues, and it will not produce recommendations that will be implemented.

An aborted or misguided assessment will have little benefit and can even make the situation worse. Prior to an assessment, the professionals generally are aware of their worst problems and often assume management is not. While this leads them to view management as mildly inept, they can assume management does not understand the issues and cannot be expected to solve them. After an assessment, this is no longer the case. A team of experts has heard their concerns and suggestions and reported them to the site manager. After all this, a manager who does not take action will be seen as either incompetent or unconcerned with the problems. In either case, morale will suffer. Thus, an assessment should not be conducted unless management is action-oriented.

3.5. Assessment Ground Rules

A written set of ground rules helps the assessment run smoothly. For an external assessment,⁵ the site manager and the assessment team leader usually sign a written agreement covering these ground rules. Such an agreement minimizes subsequent misunderstandings and ensures agreement on the critical points, such as the following:

1. The assessment team members will keep the assessment results confidential.
2. The site manager agrees to demonstrate his or her commitment by personally participating in the opening and closing assessment meetings.
3. The site manager agrees to assign two to four in-house professionals to handle the assessment arrangements and to lead the follow-up action plan work. They will be full assessment team members.
4. The site manager commits the organization to developing and implementing appropriate action plans in response to the assessment recommendations. If an action plan is not deemed appropriate, the reasons must be explained to the assessment team.

While such an agreement is essential for external assessments, it is perhaps even more important for an in-house assessment.⁶ Without a clear statement of roles and responsibilities, the assessment team members may not call on management when they should. In any case, both the site manager and the team members should clearly understand the way the assessment is to be conducted and what they are expected to do.

⁸By this, we mean an assessment performed by a team led by people from a separate organization, such as the SEI.

⁹The assessment team is led by professionals from the same parent organization.

3.6. Implementation Risks

The greatest risk with assessment is that no significant improvement actions will be taken. Without proper management focus, some superficial effort may be made, but soon everything will revert to "business as usual." To avoid this, catalysts, such as goals and management reviews, are needed to maintain the improvement priority. Long-term goals should be established first and then sub-goals for intervening two- or three-month periods. A senior management quarterly review can then help to crystallize the plans, maintain the high-level visibility of checkpoints, and create the motivation required to get things accomplished.

Some of the other key risks and potential actions to alleviate them are as follows:

Schedule conflicts: Despite the best intentions, crises that conflict with assessment plans often arise. The most damaging of these require the site manager to miss the opening or closing meeting. While these conflicts are unfortunate, they have occurred in nearly one-third of the assessments conducted by the SEI. One effective approach is to request a substitute to speak for the site manager and arrange a private meeting to cover the issues with the site manager in person.

Inadequate support: In the few cases of inadequate management support the SEI has experienced, the assessment commitment was made at too low a management level. Often only a very senior executive can take a sufficiently long-term view. Also, even fairly high-level managers are often only responsible for portions of the software work, so they cannot provide adequate organization-wide priority. It is very difficult, if not impossible, to recover from this problem. The best way to avoid this situation is to initially deal with the executive who controls the resources for the site.

Lack of follow-through: Management changes or other high-priority issues can unintentionally reduce the focus on action plan implementation. In our experience, it has not been unusual for the site manager to change between the final assessment report and action plan completion. Occasionally the action priorities were lost; more frequently the improvement efforts were successfully maintained. The most important determinants of success were the presence of an aggressive manager to lead the change efforts, a capable process improvement staff, and a clearly stated improvement goal.

4. Conducting Software Process Assessments

4.1. Assessment Phases

Assessments, as defined by the SEI, consist of six phases [OLS89]:

- i. Selection: for SEI-assisted assessments, an organization is identified as a candidate for assessment.
2. Commitment: the organization commits to participating in the assessment.
3. Preparation: the assessment team and the organization prepare to conduct the assessment.
4. Assessment: the assessment is conducted.
5. Report: the assessment team prepares a detailed written report of its findings and recommendations and delivers a briefing to the organization's senior management.
6. Follow-up: an action plan is developed and implemented by the assessed organization.

This chapter focuses on the preparation, assessment, and report phases.

4.2. Preparing for an Assessment

Following commitment to the assessment process by the senior site executive, an assessment team is selected and trained. Team members identify projects that will be assessed and hold briefings to inform those who will be involved in the assessment.

4.2.1. Forming an Assessment Team

The assessment team leader is selected first, usually by the site manager. The leader is someone who has considerable software experience, the ability to lead small groups, and experience in making presentations to senior management. He or she should have assessment experience or should obtain advice and assistance from someone who has.

All assessment team members should be experienced software developers, and one or more should have experience in each phase of the software development process. Four to six professionals form an adequate team, although more can be included. Since larger teams cost more money and

are harder to manage, an upper limit of eight to ten participants is usually wise. Whenever possible, the assessment team members should have at least eight to ten years professional software experience and should be well respected and knowledgeable about the organization. They must all be able to deal with people in an informal and non-threatening manner and be team players. It is also essential that they be motivated to advocate and participate in software process improvement.

No one should participate in the assessment who is otherwise personally involved in reviewing, supporting, or managing the projects being assessed. For example, no assessment team member should be currently serving in an audit or review capacity for any of the projects being assessed. It is also a mistake to include a line manager over any of the projects being assessed or a manager of any of the people who will be interviewed. If the organization is large enough, it is desirable to select members who are not working directly on any of the projects being assessed.

The team members should be drawn from several groups within the organization. A few members can come from assurance or support groups, but the team must appreciate the pressures of line product development. The members can be drawn from parallel projects, local test groups, or Software Quality Assurance (SQA) groups from other locations. The local SQA people, however, should not participate on the assessment team. Since smaller organizations may have trouble finding enough people who meet all criteria, they may have to make some compromises. In doing so, they should attempt to select team members who are managers or professionals working on the projects rather than staff professionals or managers. While some staff members can help to balance an assessment team, most members should have recent development experience.

With an assessment conducted with external assistance, at least one professional from the organization being assessed should participate as a full team member. This facilitates the planning process, provides the rest of the team with background on the organization, and establishes a focal point for assessment logistics and follow-up action. Since this local member is critical to the success of the effort, the site manager should be personally involved in making the selection. We have found that, up to a limit of four or five, the more local participation the better.

4.2.2. Assessment Team Preparation

In joining the assessment team, the members agree to full participation during the training period, the assessment period, and the development of recommendations and final assessment report. Unrelated phone calls are held, all other meetings and commitments rescheduled, and members should be on time for every session. Assessments are intense efforts, and it is disruptive when a team member is consistently late for meetings or otherwise preoccupied.

Typically the team leader conducts a two- or three-day training program for the entire assessment team. Team members become familiar with the assessment process and begin building a cohesive working group. Even if some members have previously been trained, they participate in this training so that they can learn about the particular organization being assessed, contribute to assessment planning, and be full and recognized members of the new team. A typical training program includes the following:

1. The assessment schedule and objectives are outlined.
2. The assessment principles are reviewed together with the software process framework.
3. The organization's mission, its management structure, and its recent history are briefly outlined.
4. The assessment guidelines are discussed, and all team members are asked to sign the written agreement.
5. A team-building exercise is conducted to assist the group in developing an effective and mutually supportive operational relationship.
6. The detailed plan for the assessment period is developed; this plan describes the purpose of each session, the participants, and their roles.
7. Where necessary, portions of the assessment process are rehearsed until the members are comfortable with their roles.
8. The details of the assessment period are arranged. Organization members describe key projects, and the team selects the most appropriate projects to assess. Following project selection, the final arrangements are made. These involve participant selection, daily schedules, and arrangements for meeting facilities and administrative support.

4.2.3. Preparing the Site

This section describes the activities of the assessment team members prior to the start of the assessment

4.2.3.1. Spreading the Word

An important activity in preparing the site for the upcoming assessment is to publicly announce that an assessment will be conducted and take steps to ensure that the software professionals are adequately and accurately informed about the following:

1. *What* an assessment is.
2. *Why* it is being conducted, and what is expected to happen as a result.
3. *Who* will be directly involved, and what the nature of their involvement will be.
4. *When* the assessment activities will occur.

Spreading the word is important because assessments can *appear* to be very similar to audits, which may arouse distrust and suspicion. If the assessment is perceived as an audit, the success of the assessment will be significantly diminished. Assessments depend on a free flow of information about how the software process works in practice, not how it could or should have been performed, nor how it will be performed the next time around. Being open and specific about the assessment is the best way to ensure that it will be perceived for what it is: an opportunity for the software organization to examine its operations with a healthy focus on improvement.

4.2.3.2. Selecting Projects

Typically five or six projects are selected as representative samples of the organization's software process. The guiding principle for selecting projects is that they represent the mainstream software business for the organization. One effective approach is to have the organization prepare a list of candidate projects for review by the entire assessment team during team training.

A common tendency is to select either the best projects or "problem" projects. The former choice usually results from wanting the organization to look good on the assessment; this is roughly analogous to cheating in your favor when balancing your checkbook. The latter choice results from the erroneous belief that assessments make everything better. While this belief may be understandable, the practices are inappropriate and counterproductive since assessments do not fix projects. An assessment of the organization's best projects will not accurately highlight areas needing improvement. Similarly,

an assessment of projects that have special problems may not result in recommendations that will have widespread benefit. An assessment is the beginning of a change process which takes time, effort, and commitment; *it is not a quick fix.*

4.2.3.3. Selecting Functional Area Representatives

During the assessment period, time is set aside for discussions with software practitioners from selected technical areas (requirements and high-level design, code and unit test, and so on). Typically, four to six professionals are selected from across the organization for each functional area. The primary purpose of these discussions is to provide the assessment team with a practitioner's perspective on the most pressing process problems facing the organization.

A secondary objective is to enroll the leading technical opinion leaders in the improvement process by encouraging them to start thinking about how activities within the scope of their influence and control might be improved.

Given this context, each functional area representative should meet the following criteria:

1. Considered an expert in the technical area by his or her peers.
2. Assigned to, and working on, one or more mainstream projects at the site (not necessarily a project included in the assessment).
3. Considered an opinion leader within the organization.

4.2.3.4. Preparing Participants

In addition to the general awareness effort described earlier, the project managers and software practitioners who will actively participate in the assessment receive additional background information concerning the upcoming assessment and their role in it. They attend one or more briefings in which the following topics are discussed:

1. How the assessment process works and its role in the larger context of software process improvement.
2. The role of project managers and functional area representatives in the assessment process.
3. Relevant events that have already taken place.
4. The schedule of upcoming assessment activities.

Assessment participants ask questions and discuss any concerns or issues. In brief, they learn what to expect during an assessment. In a smoothly run assessment, there should be as few surprises as possible for the assessment participants.

4.3. Conducting the Assessment

Questions about the organization's software process should be prepared in advance of the actual assessment period. This assures an efficient use of time and complete coverage of key process issues. The SEI has developed a questionnaire for use in process assessments and evaluations conducted in the DoD software community [HUM87a].

The questions are generally reviewed with the project managers in an initial meeting. The responses provide an overview of process status and suggest areas for further exploration. Figure 4-1 shows the flow of activities during SEI-assisted software process assessments. Key activities are discussed in the following paragraphs.

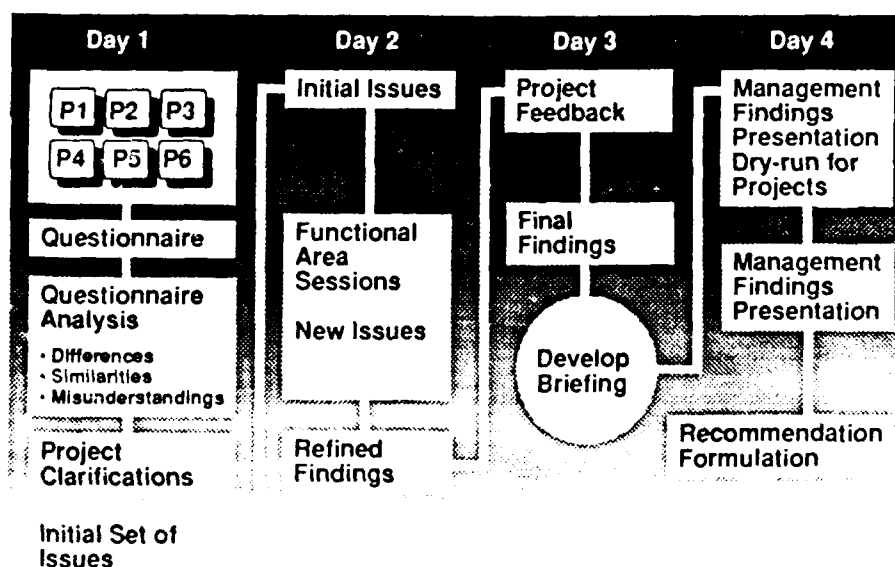


Figure 4-1: SEI On-Site Assessment Process Flow

4.3.1. Opening Assessment Briefings (Day 1)

The assessment begins with a presentation to the site manager and staff. The ground rules are discussed, as well as the assessment principles and the overall schedule. An overview meeting is then held with all the assessment participants, including the project managers and the senior professionals who will be interviewed. (Ideally, these people have also

attended the opening presentation.) Any questions and concerns are addressed at that time.

4.3.2. Reviewing Project Responses (Day 1)

The assessment team then meets in closed session to review and analyze project responses to the questions prepared in advance. The objective is to prepare the assessment team for the first round of discussions with project leaders. Some of this work can be completed in advance; this can be helpful if the assessment team is being assembled from different sites and travel funds are a problem. In addition, the advance work allows the team to rapidly focus its attention on the information at hand during the assessment. In any case, the result of this activity is a list of areas for further investigation and requests for supporting material.

4.3.3. Assessment Discussions

A significant amount of time is spent in discussions with software managers and practitioners. These discussions are necessary for the assessment team to understand the organization and to make relevant findings and meaningful action recommendations.

4.3.3.1. General Considerations

While most technical people enjoy discussing the products they are developing, this rarely provides much insight into the organization's problems. The objective of the discussions held during an assessment is to explore how projects are implemented rather than learning about the products being built. The assessment should thus focus on what the projects actually do, how they do it, the problems encountered, and the results obtained.

In conducting assessments, it can be difficult to get accurate information. The reasons include the following:

1. Questions are misunderstood. The English language is imprecise, and brief questions are invariably subject to several interpretations.
2. The respondents have different understandings of common terms. For example, discussion is often required to reach common understanding on the meaning of terms such as *high-level language*, *review*, or *environment*.
3. The respondents may not be familiar with much of the work in their own organization. Some professionals are narrowly focused on their specialty areas. Outside this sphere, they may be uninformed or even misinformed. Managers typically have a broader view, but their

hands-on experience is not always current, and their project information is often filtered by their people.

4. Occasionally, people are unwilling to risk the truth. While it is rare for someone to give misinformation, stories can generally be couched in favorable terms and valuable information can be withheld because respondents feel it does not really represent the organization's work.

As a result of these potential problems, probing and checking is an important part of every assessment, and the assessment team may ask for copies of work products. When the team members are sufficiently experienced, they can usually determine if the work was done as described.

4.3.3.2. Discussions with Project Leaders (Days 1 and 3)

Two rounds of discussions are conducted with the leaders of the projects selected for inclusion in the assessment effort. The objective of the first round of discussions is to clarify any issues identified by the assessment team during its review of project responses and to request supporting materials, if appropriate. These materials are typically documents whose existence either verifies the affirmative response to a question or describes effective practices or techniques which the assessment team feels may be more broadly applied across the organization.

The second round of discussions is used to review the supporting materials, resolve remaining issues, and review the preliminary assessment findings with the individual project managers. Only the composite findings are discussed, and the assessment team notes whether each project manager considers the findings applicable to his or her project. *If each project leader agrees that a finding is true for the organization but does not apply to his or her particular project, the assessment team needs to reevaluate the finding.*

4.3.3.3. Discussions with Functional Area Representatives (Day 2)

Meetings are also held with small groups of in-house professionals who have expertise in various facets of software development. These free-form discussions explore their views and suggestions on the key problems. These discussions should typically end with a question such as: "If you could improve one aspect of the process, what would you do and why?" In response, most groups contribute a number of creative ideas.

4.3.4. Formulating Findings (Days 2 and 3)

Following the above discussions, the assessment team meets to develop the assessment findings and produce the draft of the findings briefing to be presented to project managers on the following day. While there are many

potentially useful ways of accomplishing these tasks, they all require working under tight time constraints; typically, less than a full day is available to develop findings.

The findings are limited to no more than ten items. The team uses the following guidelines: Each finding should be a major issue for most of the projects reviewed as well as a key issue for advancing to the next maturity level. The findings must be supported by evidence from the assessment and addressable by an action recommendation. It is also important to be specific; sweeping generalizations should be avoided.

4.3.5. Presenting Findings (Day 4)

The findings briefing for senior management and assessment participants occurs on the last day of the assessment. A dry run with the participating project managers provides them an opportunity to view the presentation as it will be made to their management. The purpose is to ensure that the findings are accurate, that there are no major omissions, and that the style and terminology is appropriate.

The official findings briefing is attended by the senior site executive and staff; it is also appropriate to invite all the assessment participants. The assessment team leader or a senior team representative then briefly reviews the activities leading up to the findings and discusses each assessment finding in detail, concluding with a discussion of the recommended next steps for the organization.

4.4. Recommendations

4.4.1. Action Recommendation Considerations

Following completion of the management findings presentation, the assessment team formulates action recommendations that the organization uses as the basis for action plan development. For external assessments, the site assessment team members play an especially important role in this process. Because they are familiar with the complexities and subtleties of the organization, they are able to ensure that the recommendations are pertinent to the site's capabilities and culture.

4.4.2. The Written Assessment Report

The final assessment activity is the presentation of a written final report, including recommendations, to the site manager and staff. The recommendations highlight three or four items having the highest priority. Since no organization can handle more than a few priority tasks at a time, the total number of items requiring attention is usually limited to fewer than ten.

4.4.2.1. Role of the Report

For the following reasons, a written assessment report is always prepared:

1. Because presentations are usually tersely worded, their interpretation is highly dependent on the background and biases of the listeners. Written reports provide a less ambiguous record of what was found, what was recommended, and why. A written assessment report also provides a clear foundation for action plan preparation and implementation.
2. Writing the recommendations helps the assessment team clarify precisely what they are recommending. People who agree on a "shorthand" presentation are often surprised by the trouble they later have agreeing on a written statement of the same points.
3. The written statement constitutes the only official written record of the assessment effort. It thus provides a useful basis for future reference and comparison.

4.4.2.2. Content of the Report

A useful format for the written assessment includes the following:

1. Executive summary: briefly covers the most important aspects of the recommendations.
2. Assessment conduct: provides a written record of how the assessment was conducted, including when and where the assessment was conducted, who was on the assessment team, which projects participated, and a brief summary of the activities that took place each day.
3. Organization composite status: indicates the current maturity level at which the organization is operating, some of its most significant strengths, and a statement of goals the organization should strive for as it moves forward with process improvement.
4. Findings: provides a detailed description of the key findings, including what the team observed, specific instances of the finding (without identifying people or projects), and the implications of the finding for the organization.
5. Recommendations: briefly states the action recommendations along with supporting discussion as appropriate.

4.4.3. Recommendations Briefing

After the assessment team has completed work on the final written report, a formal briefing on the recommendations is given to the senior site executive and staff; attendance by assessment participants is suggested. Not only does this briefing present an opportunity for public discussion of the recommendations, it also serves to maintain momentum for the change process. Moreover, such management interest is another sign to the software professionals that their time was well invested and that process improvement is an important part of their job.

5. Establishing an Assessment Program

Since there are currently only a few professional assessment groups, most organizations will need to assemble an assessment team of their own.

A small staff of assessment specialists can be extremely helpful in supporting local assessment groups. If such specialists are available (through corporate or division headquarters, for example), they must also strictly observe confidentiality. The main advantage of these specialists is that they can maintain a relatively stable and repeatable assessment process. Additionally, they can help the local organizations track their progress and compare their performance with a composite of similar organizations.

In establishing an assessment program, the first step is to address scope. The program could span an entire corporation, a site location, or a single project or department. While any of these choices is possible, the first two are more likely to succeed. The software process operates in an organizational context, and it is difficult to evaluate single projects or departments without understanding the site management and support environment.

5.1. Establishing a Corporate Assessment Program

The most general case of assessment spans the software work of an entire corporation. Although it may be practical to set up assessment teams at a local level, there are some special considerations that can best be discussed after reviewing the establishment of a corporate effort.

The first step in forming a corporate assessment is the decision by management to do so. This decision requires a senior manager who is convinced that software assessments are desirable and who has the resources and authority to initiate and sustain them. Typically this manager is the corporate staff executive for software or for quality. If no such executive exists, the corporate vice president for engineering is an appropriate alternate. In any case, this executive must consider an assessment program important and agree to implement it.

Next, the executive names an individual to do the planning and recruit the assessment staff. This person should be a seasoned software executive with experience managing people and a demonstrated ability to operate effectively in the corporate staff environment. These qualities are essential because of the confidential nature of assessments and the difficulty of convincing site managers that a corporate staff will review their operations without reporting to corporate headquarters. Conversely, it is often difficult to convince skeptical corporate managers that the information gathered in assessments should be withheld from them. This problem is particularly

severe when corporate software cost and schedule performance has been poor.

5.2. Planning a Corporate Assessment Program

The assessment program leader's initial job is to define the charter and the organizational scope, identify staffing needs, and produce a proposed operating plan. This work should initially be reviewed and approved by the responsible corporate staff head and then discussed with the line managers and site directors whose organizations will be assessed. These discussions should not only seek agreement with the overall approach but should also discuss staffing needs and operational methods. The meetings should always start with an unequivocal statement of the confidentiality provisions. Line management will not willingly participate in assessments that are not confidential. They also will not generally understand or believe the confidentiality provisions when they are first described. Confidentiality, therefore, should be described at the opening and close of the meetings and frequently reinforced in the intervening discussions.

5.3. Staffing for Corporate Assessment Groups

Staffing is handled at three levels: permanent, rotating, and assessment team membership. While the permanent staff should be kept small, it must include at least three to five professionals. A smaller group will not build an adequate experience base to provide leadership to the effort. Since assessments are hard work and can be very tiring, it is also important to have a large enough staff to permit the members to participate in alternate assessments. Ideally, after the organization is fully operational, two assessment teams should handle alternate assessments.

In addition to the permanent staff, it is also important to have rotating members. These members are temporary site assignees who participate for one- to two-year periods. Since recruiting and training take time, one year is the recommended minimum. Even though recruiting personnel for longer than one-year assignments is more difficult, 18 months should be the normal assignment period if at all possible.

The reasons for using rotational assignments to staff the corporate group are three: staff quality, training, and enrollment. The laboratories will be reluctant to provide good people to corporate headquarters for permanent assignments. Generally they will not do so at all unless a corporate executive development program provides an incentive [HUM87b]. By participating in the assessment group, staff members become trained in assessment methods and are exposed to the key problems of software process improvement. After participating in several assessments, experienced software professionals have a better appreciation of the need for software process improvement. By witnessing the frequent struggles of

software groups with previously solved problems, they will better appreciate the costs of a low-maturity software process. Their new knowledge serves to enroll them in the process improvement effort. After exposure to several assessments, professionals often become enthusiastic agents for software process change. Thus they become a critical resource.

The third staffing need is for location members on each assessment team. Once a site has agreed to participate in an assessment, however, this is not difficult to arrange.

5.4. Initiating Corporate Assessments

When the plan is approved and staffing complete, the first assessment can begin. It starts with a training program for the entire assessment team. In the absence of trained people to lead such sessions, a training plan and teaching materials should be developed by the staff. Development can be done with the aid of publicly available materials [CON73, HUM87a, HUM87b, HUM88, HUS75, RAD85, ROD78]. After the plan has been developed, the entire assessment team takes the course together before the first assessment. During this training and after the assessment, the team should critique the course and the assessment methods and suggest areas for improvement.

5.5. Establishing a Site Assessment Program

In establishing the assessment effort at a site level, the general approach is similar to that described for the corporate plan. The key differences are that the activity will be on a smaller scale and that local management support is even more important. A local assessment effort should be facilitated by a software engineering process group (SEPG)⁷ [HUM88, HUM89]. The SEPG manager typically leads the assessments, and several SEPG staff members participate.

Site assessments are conducted every 18 months to 2 years, so a full-time assessment staff is not needed. Additionally, all the SEPG members should be given experience with the assessment process. The assessment team can thus be somewhat larger to give all members this exposure. Assessment should be a scheduled part of the SEPG plan so that other work is not adversely affected.

¹⁰An SEPG is a group of software professionals specifically chartered to focus on software process improvement. See Section 2.6.

5.6. Other Considerations

Three considerations deserve special emphasis: corporate support, local management support, and assessment credibility.

Support from a high level of management is essential for a successful assessment effort (the higher the level, the better). Typically the most serious problems with the software process can only be addressed by management, and often these solutions require resource commitments. In many organizations, even site managers can staff only minor activities without support and assistance from headquarters. Senior management also needs to be aware of the assessment activity and fully support it. Otherwise, they may not give its recommendations sufficient priority. The first time a senior executive says, "I don't care about the assessment; what I want you to do is this..." the improvement program is over.

Local management support is essential also. If the site manager and project managers do not support the effort, an organization should not attempt an assessment. There is no point in rushing in to do an assessment if local managers are not on board, even if corporate management wants it done immediately. The local managers are the ones who must take action following the assessment. If they do not support the effort, nothing will get done. Further, if they do not agree to support the assessment, it will be unlikely to find the key problems. Before proceeding, be sure that local managers support the assessment. If they do not, find out why and resolve the problems. Then proceed.

Regarding credibility, software professionals and managers are properly skeptical about new approaches to their problems. They may have seen previous schemes that had little or no positive impact on their jobs, and they may doubt that an assessment can help. Therefore, it is important to consciously build on prior work and to cite that work in the preparation and conduct of the assessment. A framework that builds on prior experience demonstrates thoughtful preparation and lends credibility to the effort.

6. Conclusions

As organizations recognize the need to improve their software capability, they will find the assessment process increasingly important. It provides an orderly identification of the most critical problems and helps initiate a comprehensive improvement effort. Its most important single benefit, however, is to expose the management and technical professionals to the need for continually improving the way they do their work. That, of course, is the ultimate objective: to establish a dynamic process that evolves in step with the needs and capabilities of the people who use it.

References

1. [AGR81] Agresti, W.W., "Applying Industrial Engineering to the Software Development Process," *Proceedings, IEEE Fall COMPCON*, Washington, DC: IEEE Computer Society Press, 1981, pp. 264-270.
2. [BAU75] Bauer, F. (ed.), *Software Engineering - An Advanced Course*, Springer-Verlag, 1975.
3. [BRO87] Brooks, F., "No Silver Bullet, Essence and Accidents of Software Engineering", *IEEE Computer*, April 1987.
4. [CAR87] Card, D.N., Clark, T.L., and Berg, R.A., "Improving Software Quality and Productivity," *Information and Software Technology*, vol. 29, no. 5, June 1987.
5. [CON73] The Conference Board, Conference Report 605, "Organizational Development: A Reconnaissance," New York, 1973.
6. [CRO79] Crosby, P.B., *Quality is Free - The Art of Making Quality Certain*, McGraw-Hill, 1979.
7. [CUR87] Curtis, W., Krasner, H., Shen, V. and Iscoe, N., "On Building Software Process Models Under the Lamppost," *IEEE Proceedings, 9th International Conference on Software Engineering*, Monterey, CA, March 30 to April 2, 1987.
8. [CUR88] Curtis, B., Krasner, H., Iscoe, N., "A Field Study of the Software Design Process for Large Systems," *Communications of the ACM*, November 1988.
9. [DEM82] Deming, W.E., *Out of the Crisis*. Cambridge, MA: MIT Center for Advanced Engineering Study, 1982.
10. [GAR65] Gardner, J., *Renewal of Organizations*, 20th Annual Meeting of the Board of Trustees, Midwest Research Institute, Kansas City, MO, May 3, 1965.
11. [HUM87a] Humphrey, W.S., Sweet, W. et al., *A Method for Assessing the Software Engineering Capability of Contractors*, Software Engineering Institute, CMU/SEI-87-TR-23; DTIC: ADA 187230, September, 1987.

12. [HUM87b] Humphrey, W. S., *Managing for Innovation - Leading Technical People*, Englewood Cliffs, NJ: Prentice-Hall, 1987.
13. [HUM88] Humphrey, W.S., "Characterizing the Software Process: A Maturity Framework," *IEEE Software Magazine*, March 1988.
14. [HUM89] Humphrey, W.S., Kitson, D.H., and Kasse, T.C., *The State of Software Engineering Practice: A Preliminary Report*, Software Engineering Institute, (CMU/SEI-89-TR-1), February 1989.
15. [HUS75] Huse, E. H., *Organization Development and Change*, St. Paul, MN: West Publishing Company, 1975.
16. [OLS89] Olson, T. H., Humphrey, W.S., and Kitson, D. H., *Conducting SEI-Assisted Software Process Assessments*, Software Engineering Institute, (CMU/SEI-89-TR-7), February 1989.
17. [RAD85] Radice, R. A., Harding, J. T., Munnis, P. E., and Phillips, R. W., "A Programming Process Study," *IBM Systems Journal*, vol. 24, no. 2, 1985.
18. [ROD78] Rodgers, David, *Can Business Management Save the Cities*, New York: MacMillan Publishing Co., Inc., Free Press, 1978.
19. [WAL86] Walton, Mary, *The Deming Management Method*, New York: Putnam Publishing Group, 1986.
20. [ZAN82] Zand, Dale, Peter Drucker Management Conference, New York City, April 22, 1982.